The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.

TLP: WHITE

Traffic Light Protocol (TLP): WHITE information may be distributed without restriction, subject to copyright controls.

http://www.us-cert.gov/tlp/

DATE(S) ISSUED:
06/06/2016

SUBJECT:
Vulnerability in WordPress Mobile Detector Plugin Could Allow for Remote Code Execution

OVERVIEW:
A vulnerability in the WordPress Mobile Detector plugin has been discovered, which could allow for remote code execution. WordPress Mobile Detector is used to display content on WordPress sites in a format suitable for phones and tablet devices. Successful exploitation of this vulnerability could result in an attacker being able to execute remote code in the context of the web server process or could allow for the uploading of arbitrary files. This may permit an attacker access to sensitive information and compromise the system.

THREAT INTELLIGENCE:
There are reports of this vulnerability being actively exploited in the wild.

SYSTEM AFFECTED:
WordPress WP Mobile Detector plugins prior to version 3.6

RISK:
Government:

- Large and medium government entities: **High**
- Small government entities: **High**

Businesses:

- Large and medium business entities: **High**
- Small business entities: **High**

Home users: Low

TECHNICAL SUMMARY:
WordPress Mobile Detector is prone to a vulnerability that could allow for remote code execution due to a failure to sanitize user-supplied input submitted to the 'src' parameter of the 'resize.php' file located in the plugin directory. An attacker can make a POST request to the php file with a malicious URL as the payload to be uploaded onto the plugin cache directory, using the following syntax:

http://[site]/wp-content/plugins/wp-mobile-detector/resize.php?src=[URL of upload file]

After uploading a malicious file (such as a php file that can execute remote code on the server), an attacker can execute his malicious payload with a call to the uploaded script on the server.

This attack utilizes the file_get_contents() function to upload a file onto the server and would require the allow_url_fopen field of the PHP configuration to be enabled for the function to upload a file whose path is given by a URL. Hence, for this attack to work, allow_url_fopen needs to be enabled.

Successful exploitation of this vulnerability could result in an attacker being able to execute remote code in the context of the web server process or could allow for the uploading of arbitrary files. This may allow an attacker access to sensitive information and compromise the system.

RECOMMENDATIONS:
The following actions should be taken:

- Remove the Mobile Detector plugin if it is not needed.
- Update the Mobile Detector plugin to its most recent version.
- Otherwise, disable the allow_url_fopen of the PHP configuration if that field is not needed. Note: This may render your WordPress installation inoperable or unstable.
- Consider implementation of a Web Application Firewall to mitigate common threats to publically available webservers.
- Review and follow WordPress hardening guidelines - http://codex.wordpress.org/Hardening_WordPress
- Confirm that the operating system and all other applications on the system running this CMS are updated with the most recent patches.
- Deploy NIDS to detect and block attacks and anomalous activity such as crafted requests containing suspicious URI sequences.
- Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack.

REFERENCES:
WordPress Mobile Detector:
https://wordpress.org/plugins/wp-mobile-detector/
https://wordpress.org/plugins/wp-mobile-detector/changelog/

Sucuri:
https://blog.sucuri.net/2016/06/wp-mobile-detector-vulnerability-being-exploited-in-the-wild.html

Plugin Vulnerabilities:
https://www.pluginvulnerabilities.com/2016/05/31/aribitrary-file-upload-vulnerability-in-wp-mobile-detector/